

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

日本国特許庁
PATENT OFFICE
JAPANESE GOVERNMENT

K. Kikuchi
5/8/01
Q 64347
10f1

JCP86 U.S. PTO
09/849983
05/08/01

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日

Date of Application:

2000年 5月31日

出願番号

Application Number:

特願2000-161351

出願人

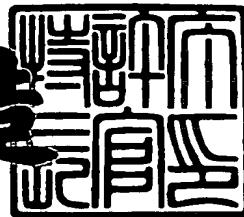
Applicant(s):

日本電気株式会社

2001年 3月 2日

特許庁長官
Commissioner,
Patent Office

及川耕造



出証番号 出証特2001-3012492

【書類名】 特許願
【整理番号】 65000543
【提出日】 平成12年 5月31日
【あて先】 特許庁長官殿
【国際特許分類】 G06F 15/00
【発明者】
【住所又は居所】 東京都港区芝五丁目7番1号 日本電気株式会社内
【氏名】 菊池 謙一
【特許出願人】
【識別番号】 000004237
【氏名又は名称】 日本電気株式会社
【代理人】
【識別番号】 100088812
【弁理士】
【氏名又は名称】 ▲柳▼川 信
【手数料の表示】
【予納台帳番号】 030982
【納付金額】 21,000円
【提出物件の目録】
【物件名】 明細書 1
【物件名】 図面 1
【物件名】 要約書 1
【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 マルチプログラム処理装置及びそれに用いるファイルのロールバック方式

【特許請求の範囲】

【請求項1】 複数のプログラムが同時に動作可能なマルチプログラム処理装置であつて、前記複数のプログラムによる参照・更新の対象となるファイルをブロック単位に排他制御を行う排他制御手段と、前記プログラムで更新中のブロックに対応する更新前のブロックを一時保持する保持手段とを有し、あるプログラムでデッドロック及び異常終了のいずれかが発生してそれまでの更新をロールバックする間に前記保持手段に一時保持したブロックを用いて他のプログラムからの当該ロールバックの対象となるブロックへの参照・更新を可能とするようにしたことを特徴とするマルチプログラム処理装置。

【請求項2】 前記ファイルのブロック毎に当該ブロックを利用中のプログラムを特定するプログラム識別情報と、前記ロールバック中か否かを示す情報とを保持するブロック状態表を含み、前記排他制御手段は前記ブロック状態表の記録内容に応じて前記排他制御を行うようにしたことを特徴とする請求項1記載のマルチプログラム処理装置。

【請求項3】 前記プログラムからの参照・更新要求において前記デッドロック及び前記異常終了のいずれかが発生した時に前記保持手段に一時保持した更新前のブロックをそれに対応するファイルに書戻して更新を取り消すロールバック手段を含むことを特徴とする請求項1または請求項2記載のマルチプログラム処理装置。

【請求項4】 前記ロールバック手段は、ブロックが他のプログラムによって更新されていれば当該ブロックに対するロールバックを不要としつつ前記他のプログラムによって更新されていなければ当該ブロックに対するロールバックを行うようにしたことを特徴とする請求項1から請求項3のいずれか記載のマルチプログラム処理装置。

【請求項5】 複数のプログラムが同時に動作可能なマルチプログラム処理

装置のファイルのロールバック方式であって、前記複数のプログラムによる参照・更新の対象となるファイルをブロック単位に排他制御を行い、あるプログラムでデッドロック及び異常終了のいずれかが発生してそれまでの更新をロールバックする間に、前記プログラムで更新中のブロックに対応する更新前のブロックを一時保持する保持手段に一時保持したブロックを用いて他のプログラムからの当該ロールバックの対象となるブロックへの参照・更新を可能とするようにしたことを特徴とするファイルのロールバック方式。

【請求項6】 前記ファイルのブロック毎に当該ブロックを利用中のプログラムを特定するプログラム識別情報と、前記ロールバック中か否かを示す情報を保持するブロック状態表の記録内容に応じて前記排他制御を行うようにしたことを特徴とする請求項5記載のファイルのロールバック方式。

【請求項7】 前記プログラムからの参照・更新要求において前記デッドロック及び前記異常終了のいずれかが発生した時に前記保持手段に一時保持した更新前のブロックをそれに対応するファイルに書き戻して更新を取り消すようにしたことを特徴とする請求項5または請求項6記載のファイルのロールバック方式。

【請求項8】 ブロックが他のプログラムによって更新されていれば当該ブロックに対するロールバックを不要としつつ前記他のプログラムによって更新されていなければ当該ブロックに対するロールバックを行うようにしたことを特徴とする請求項5から請求項7のいずれか記載のファイルのロールバック方式。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明はマルチプログラム処理装置及びそれに用いるファイルのロールバック方式に関し、特に複数のプログラムが同時に動作可能なマルチプログラム処理装置におけるファイルのロールバック方式に関する。

【0002】

【従来の技術】

従来、マルチプログラム処理装置においては、複数のプログラムが同時に動作可能となっており、一つのファイルを複数のプログラムで同時に参照・更新する

際に、複数のプログラムからの参照・更新に対して重要なファイルを保護する機能として、ブロック排他及びロールバックがある。

【0003】

ブロック排他とはあるプログラムが参照・更新しているファイルに対して、あるプログラムによる参照・更新が終了するまで、そのファイルが他のプログラムによって参照・更新されるのを抑止する処理である。

【0004】

また、ロールバックとはあるプログラムが動作中に何らかの異常が発生し、正常に処理を完了できない場合、つまりデッドロックや異常終了が発生した場合、そのプログラムの動作に関連する動作を中止し、それに関連する情報を動作前の状態に戻す処理である。

【0005】

【発明が解決しようとする課題】

しかしながら、上述した従来のファイル保護機能では、どのプログラムもブロック排他において均等に扱われるため、デッドロックまたはプログラムの異常終了によってロールバックが行われる際に、次に必要とするブロックがあってもそれがロールバック対象のブロックである場合、ロールバック終了までそのブロックを参照・更新することができない。

【0006】

そこで、本発明の目的は上記の問題点を解消し、ロールバックの対象となるブロックの参照・更新をその終了を待たずに優先して行うことができ、一つのファイルを複数のプログラムで同時に参照・更新する際のプログラムの処理効率を向上させることができるマルチプログラム処理装置及びそれに用いるファイルのロールバック方式を提供することにある。

【0007】

【課題を解決するための手段】

本発明によるマルチプログラム処理装置は、複数のプログラムが同時に動作可能なマルチプログラム処理装置であって、前記複数のプログラムによる参照・更新の対象となるファイルをブロック単位に排他制御を行う排他制御手段と、前記

プログラムで更新中のブロックに対応する更新前のブロックを一時保持する保持手段とを備え、あるプログラムでデッドロック及び異常終了のいずれかが発生してそれまでの更新をロールバックする間に前記保持手段に一時保持したブロックを用いて他のプログラムからの当該ロールバックの対象となるブロックへの参照・更新を可能とするようにしている。

【0008】

本発明によるファイルのロールバック方式は、複数のプログラムが同時に動作可能なマルチプログラム処理装置のファイルのロールバック方式であって、前記複数のプログラムによる参照・更新の対象となるファイルをブロック単位に排他制御を行い、あるプログラムでデッドロック及び異常終了のいずれかが発生してそれまでの更新をロールバックする間に、前記プログラムで更新中のブロックに対応する更新前のブロックを一時保持する保持手段に一時保持したブロックを用いて他のプログラムからの当該ロールバックの対象となるブロックへの参照・更新を可能とするようにしている。

【0009】

すなわち、本発明のファイルのロールバック方式は、一つのファイルをブロック単位に排他制御を行って複数のプログラムで同時に参照・更新する際に、あるプログラムでデッドロックまたは異常終了が発生してそれまでの更新をロールバックする間にも、他のプログラムからロールバック対象となるブロックの参照・更新を可能とする手段を提供するものである。

【0010】

より具体的に、本発明の前提となる基本的な動作から説明する。本発明のファイルI/O手段はプログラムからのデータファイルに対する参照・更新要求を処理する。また、ファイルI/O手段ではブロック排他手段に対してプログラム間で参照・更新対象となるブロックが利用可能であるか、またデッドロックが発生しないかを照会及び利用状況の登録を要求する。

【0011】

ブロック排他手段では要求に応じてブロック状態表を参照・更新する。ブロックが利用可能であり、プログラムからの要求が参照の場合、ファイルI/O手段

でデータファイルから目的のブロックを読み込んでプログラムに返却する。また、要求が更新の場合、ファイルI/O手段ではジャーナル収集手段に対し、更新前のブロックをデータファイルからジャーナルファイルにジャーナルとして保存させ、その後にデータファイルを更新する。

【0012】

プログラムからの参照・更新要求においてデッドロックが発生したり、プログラムが異常終了した場合、ロールバック手段がそれまでに蓄えられたジャーナルをジャーナルファイルからデータファイルに書戻すことによって、更新を取り消す。

【0013】

次に、本発明における動作を説明する。本発明においては以下のようにしてロールバックする間にも、他のプログラムに対して対象ブロックの参照・更新を可能とする。

【0014】

プログラムBでデッドロックまたは異常終了が発生した場合、ロールバック手段ではブロック排他手段に対し、プログラムBが参照・更新したブロックにロールバック中である印を付けさせた後に、ロールバックを開始する。

【0015】

プログラムAからデータファイルに対する参照が要求された場合、ファイルI/O手段ではブロック排他手段に当該ブロックを照会する。当該ブロックがプログラムBで更新されていなければ、上記の基本的な動作と同じになるが、更新されていた場合に、ファイルI/O手段はデータファイルから当該ブロックを読み込む代わりにジャーナル共用手段に対し、当該ブロックに係るジャーナルを要求し、プログラムBが更新する前のブロックをジャーナルファイルから取得してプログラムAに渡す。

【0016】

また、プログラムAからデータファイルに対する更新が要求された場合、ファイルI/O手段でブロック排他手段に当該ブロックを照会した結果、当該ブロックがプログラムBで更新されていなければ、やはり上記の基本的な動作と同じに

なるが、プログラムBで更新されていたとすると、当該ブロックの更新はプログラムAによるものと印し直して、プログラムBのロールバックを不要とする。

【0017】

この後、ジャーナル収集手段が更新前のブロックをデータファイルから読込む代わりに、ジャーナル共用手段でジャーナルファイルから当該ブロックに係るジャーナルを取得し、プログラムAによる更新としてジャーナルファイルに保存する。

【0018】

一方、ロールバック手段では、ジャーナルファイルに蓄えられたジャーナルをデータファイルに書戻す訳であるが、ブロック排他手段に当該ブロックが他のプログラムによって更新されたか否かを照会する。他のプログラムが更新していれば、当該ブロックに対するロールバックが不要となり、更新されていなければ、当該ブロックに対するロールバックを行う。

【0019】

上記のように、一つのファイルをブロック単位に排他制御を行うことで、デッドロックまたはプログラムの異常終了によってロールバックが行われる際、ロールバックの対象となるブロックの参照・更新をその終了を待たずに優先して行うことが可能となり、一つのファイルを複数のプログラムで同時に参照・更新する際にプログラムの処理効率を向上させることが可能となる。

【0020】

【発明の実施の形態】

次に、本発明の一実施例について図面を参照して説明する。図1は本発明の一実施例によるマルチプログラム処理装置の構成を示すブロック図である。図1において、本発明の一実施例によるマルチプログラム処理装置はファイルI/O（入出力）手段1と、ブロック排他手段2と、ブロック状態表3と、ジャーナル共用手段4と、ジャーナル収集手段5と、ジャーナルファイル6と、ロールバック手段7と、データファイル8とを含んで構成され、複数のプログラムが同時に動作可能となっている。

【0021】

ブロック状態表3はブロック排他手段2がデータファイル毎のファイル識別情報と、ブロック毎のファイル内相対ブロック番号、プログラム識別情報、ロールバック中か否かを示す印とを記録する。

【0022】

ファイルI/O手段1はプログラムからの参照要求に対し、まず当該ブロックの状態をブロック排他手段2に照会する。ブロック排他手段2ではブロック状態表3上の当該ブロックが参照・更新されているか、また他のプログラムで更新されていても、ロールバック中の印があるかどうかを確認する。

【0023】

ブロック排他手段2は当該ブロックが更新されていなければ、ファイル内相対ブロック番号とプログラム識別情報とを格納し、ファイルI/O手段1でデータファイル8からブロックを読み込んでプログラムに渡す。ブロック排他手段2は当該ブロックが他のプログラムで更新中ならば、プログラムの要求を待たせる。

【0024】

その結果、デッドロックが発生するようであれば、ブロック排他手段2はプログラムの要求を拒否する。尚、プログラムの要求を待たせたり、デッドロックを検出する処理の詳細な動作に関しては本発明の説明の範囲外であるので、その説明については省略する。

【0025】

当該ブロックが他のプログラムで更新中であっても、ロールバック中であれば、ファイルI/O手段1ではデータファイル8から当該ブロックを読み込む代わりに、ジャーナル共用手段4に対して当該ブロックに係るジャーナルを要求し、当該ブロックの内容を他のプログラムが更新する前の状態でプログラムに渡す。

【0026】

プログラムからの更新要求に対しても、上記と同様に、ファイルI/O手段1はブロック排他手段2にブロックの状態を照会する。他のプログラムが参照も更新もしていなければ、ファイルI/O手段1はジャーナル収集手段5によって更新前のブロックをデータファイル8からジャーナルファイル6にジャーナルとして保存させ、その後、データファイル8を更新する。

【0027】

他のプログラムが更新中であっても、ロールバック中であれば、ファイルI/O手段1はプログラム識別情報を要求元プログラムのものと入れ替え、ジャーナル共用手段4で更新前のブロックをジャーナルファイル6から取得して新たなプログラムからの更新として保存し、データファイル8を更新する。尚、プログラム識別情報を入れ替えることによって、ロールバック手段7は後述するように、当該ブロックのロールバックが不要であると認識する。

【0028】

ブロック排他手段2でデッドロックが検出されたり、プログラムが異常終了した場合には、ロールバック手段7がデータファイルのロールバックを行うが、その際、ロールバック手段7ではまず、ブロック排他手段2に該当プログラムが更新したブロックにロールバック中の印を付けさせる。

【0029】

続いて、ロールバック手段7ではジャーナルファイル6からジャーナルを取得してきて、該当ブロック毎にブロック排他手段2に照会し、既に他のプログラムが更新しているかどうかを判定する。ロールバック手段7は更新されていなければロールバックし、更新されていればロールバックを不要とする。

【0030】

図2及び図3は本発明の一実施例によるマルチプログラム処理装置の動作を示すフローチャートである。これら図1～図3を参照して本発明の一実施例によるマルチプログラム処理装置の動作について説明する。

【0031】

プログラムAがデータファイル8のI/Oを行おうとする場合、ファイルI/O手段1はそれが参照か更新かを判定する(図2ステップS1)。参照であれば、ブロック排他手段2がブロック状態表3を照会し、当該ブロックが例えばプログラムBによって更新中かどうかを判定する(図2ステップS2)。

【0032】

当該ブロックがプログラムBによって更新中でなければ、ブロック排他手段2はプログラムAの情報をブロック状態表3に登録し(図2ステップS3)、ファ

イルI/O手段1がデータファイル8から当該ブロックを読み込んで（図2ステップS4）、プログラムAに返却する。

【0033】

当該ブロックがプログラムBによって更新中の場合、ブロック排他手段2はさらにロールバック中かどうかを判定する（図2ステップS5）。ロールバック中でなければ、ブロック排他手段2は今回の参照要求を待たせるか、デッドロックを検出してエラーを返却することになる（図2ステップS17, S18）。

【0034】

ロールバック中であれば、上記と同様に、ブロック排他手段2はプログラムAの情報をブロック状態表3に登録し（図2ステップS6）、ジャーナル共用手段4がジャーナルファイル6からプログラムBが更新する前の状態のブロックを読み込んで（図2ステップS7）、プログラムAに返却する。

【0035】

また、プログラムAからの要求が更新であれば、ブロック排他手段2はブロック状態表3を照会することによって、当該ブロックがプログラムBによって参照・更新中でないかどうかを判定する（図2ステップS8）。

【0036】

上記のいずれでもなければ、ブロック排他手段2はプログラムAの情報をブロック状態表3に登録し（図2ステップS9）、ジャーナル収集手段5が更新前のブロックをデータファイル8から読み込んでジャーナルファイル6に保存した後（図2ステップS10, S11）、ファイルI/O手段1がデータファイル8に新しいブロックを書き出す（図2ステップS12）。

【0037】

当該ブロックがプログラムBによって更新中でなければ、つまり参照中であれば（図2ステップS13）、ブロック排他手段2は今回の更新要求を待たせるか、デッドロックを検出してエラーを返却する（図2ステップS17, S18）。

【0038】

当該ブロックがプログラムBによって更新中の場合、ブロック排他手段2はさらにロールバック中かどうかを判定する（図2ステップS14）。ロールバック

中でなければ、ブロック排他手段2は今回の更新要求を待たせるか、デッドロックを検出してエラーを返却する（図2ステップS17, S18）。

【0039】

ロールバック中であれば、ブロック排他手段2は当該ブロックを更新中のプログラムがプログラムAであるとしてブロック状態表3に登録し直し（図2ステップS15）、ジャーナル共用手段4がプログラムBが更新する前の状態のブロックをプログラムAによるものとしてジャーナルファイル6に登録し直す（図2ステップS16）。その後、上記と同様に、ファイルI/O手段1がデータファイル8に新しいブロックを書き出す（図2ステップS12）。

【0040】

一方、プログラムBでデッドロックまたは異常終了が発生した場合、ロールバック手段7がブロック排他手段2に対し、ブロック状態表3に更新したブロックにロールバック中の印を付けさせる（図3ステップS21）。この後、ロールバック手段7ではジャーナルファイル6から更新前のブロックであるジャーナルを読み込んでデータファイル8に書き戻すが（図3ステップS22, S24）、先にロールバック中の印を付けたことによって、当該ブロックが既にプログラムAによって更新される場合がある。

【0041】

この場合、当該ブロックのロールバックを行う必要がない（図3ステップS23）。いずれにしてもプログラムBの更新分のジャーナルを全て処理し（図3ステップS25）、その後、ブロック排他手段2がブロック状態表3からプログラムBの情報を削除する（図3ステップS26）。

【0042】

図4は図1のブロック状態表3の構成例を示す図である。図4において、ブロック状態表3にはファイル毎のファイル識別情報と、ブロック毎の相対ブロック番号、参照及び更新プログラム情報、ロールバック中であるかどうかの印とが格納されている。

【0043】

図5（a）, (b) は本発明の一実施例によるマルチプログラム処理装置の動

作を示す図である。これら図4及び図5を参照して本発明の一実施例によるマルチプログラム処理装置の動作について具体的に説明する。

【0044】

他にデータファイル8を参照・更新中のプログラムがない状態からプログラムAがデータファイル8のブロック#1を参照し、ブロック#2を更新すると、ブロック状態表3は図5(a)に示すようになる(図2ステップS1~S4, S8~S12)。尚、図5(a), (b)においては設定されない箇所の情報を点線で表記している。

【0045】

続いて、プログラムBがブロック#x, #yを更新すると、図5(b)に示すようになる。この状態から、プログラムBが異常終了し、ロールバックが開始されると、プログラムBが更新したブロックにロールバック中の印が付けられ(図3ステップS21)、ブロック状態表3は図5(b)に示すようになる。

【0046】

さらに、プログラムAがブロック#xを参照しようとすると、ブロック#xはロールバック中であると印されているので、プログラムAの参照は認められる(図2ステップS5, S6)。この時、ブロック#xはデータファイル8ではなく、ジャーナルファイル6からプログラムBが更新する前の状態で読込まれる(図2ステップS7)。

【0047】

また、プログラムAがブロック#yを更新しようとする場合にも、やはりロールバック中であると印されているので、プログラムAの更新が認められる(図2ステップS13~S15)。この時、更新前のブロックはデータファイル8から取得するのではなく、ジャーナルファイル6から取得される(図2ステップS16)。図5(b)の一番右側の状態はプログラムAがブロック#xを参照し、ブロック#yを更新した時の状態を示している。

【0048】

プログラムBに対するロールバックを行う際には、ブロック#xについてロールバックが行われるが、ブロック#yについては行われない(図3ステップS2)

2～S25)。

【0049】

このように、一つのファイルに対してブロック単位に排他制御を行うことによって、デッドロックまたはプログラムの異常終了によってロールバックが行われる際、ロールバックの対象となるブロックの参照・更新をその終了を待たずに優先して行うことができ、一つのファイルを複数のプログラムで同時に参照・更新する際にプログラムの処理効率を向上させることができる。

【0050】

【発明の効果】

以上説明したように本発明によれば、複数のプログラムが同時に動作可能なマルチプログラム処理装置において、複数のプログラムによる参照・更新の対象となるファイルをブロック単位に排他制御を行い、あるプログラムでデッドロック及び異常終了のいずれかが発生してそれまでの更新をロールバックする間に、プログラムで更新中のブロックに対応する更新前のブロックを一時保持する保持手段に一時保持したブロックを用いて他のプログラムからの当該ロールバックの対象となるブロックへの参照・更新を可能とすることによって、ロールバックの対象となるブロックの参照・更新をその終了を待たずに優先して行うことができ、一つのファイルを複数のプログラムで同時に参照・更新する際のプログラムの処理効率を向上させることができるという効果がある。

【図面の簡単な説明】

【図1】

本発明の一実施例によるマルチプログラム処理装置の構成を示すブロック図である。

【図2】

本発明の一実施例によるマルチプログラム処理装置の動作を示すフローチャートである。

【図3】

本発明の一実施例によるマルチプログラム処理装置の動作を示すフローチャートである。

【図4】

図1のブロック状態表の構成例を示す図である。

【図5】

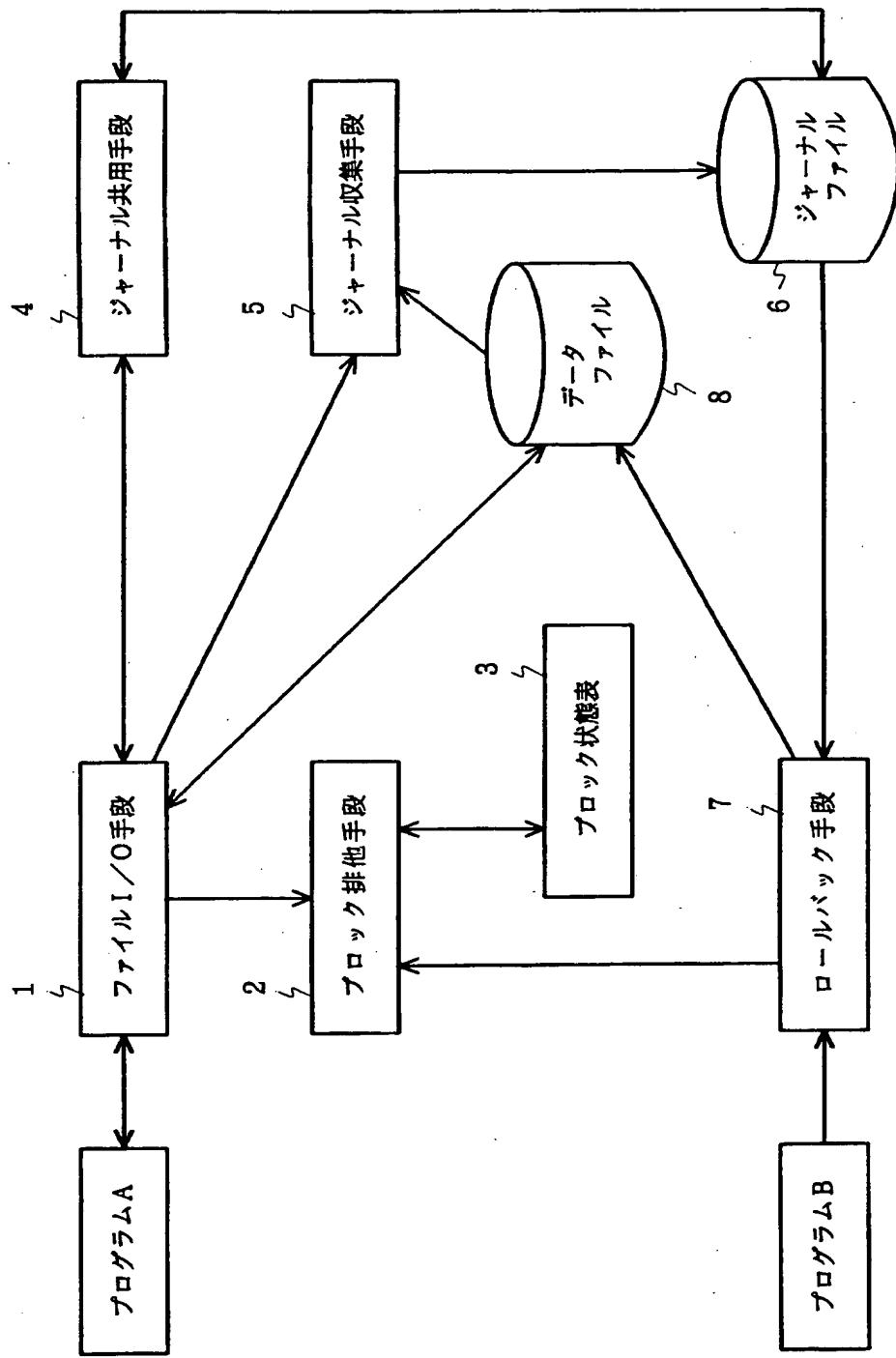
(a), (b) は本発明の一実施例によるマルチプログラム処理装置の動作を示す図である。

【符号の説明】

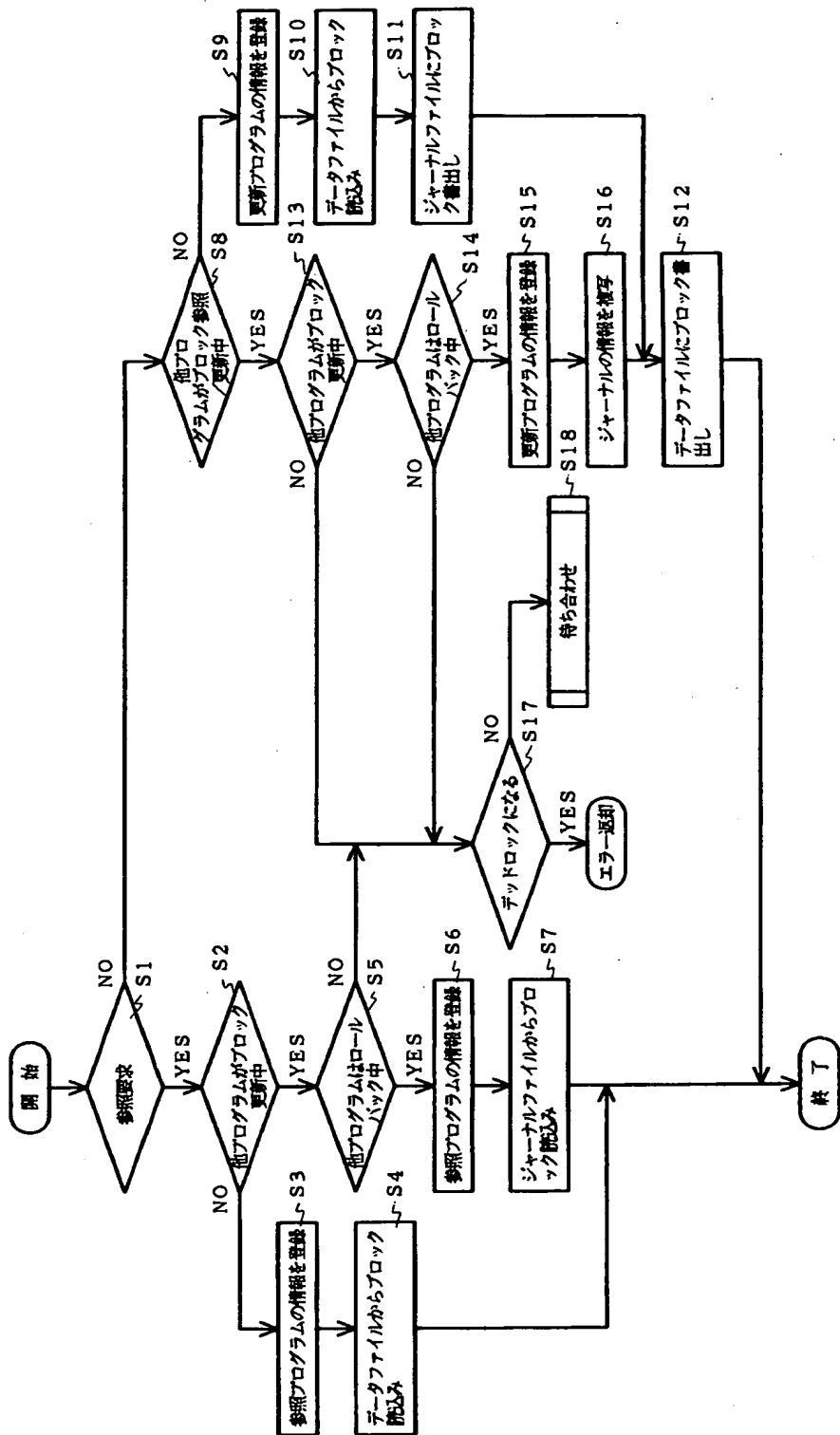
- 1 ファイルI/O手段
- 2 ブロック排他手段
- 3 ブロック状態表
- 4 ジャーナル共用手段
- 5 ジャーナル収集手段
- 6 ジャーナルファイル
- 7 ロールバック手段
- 8 データファイル

【書類名】 図面

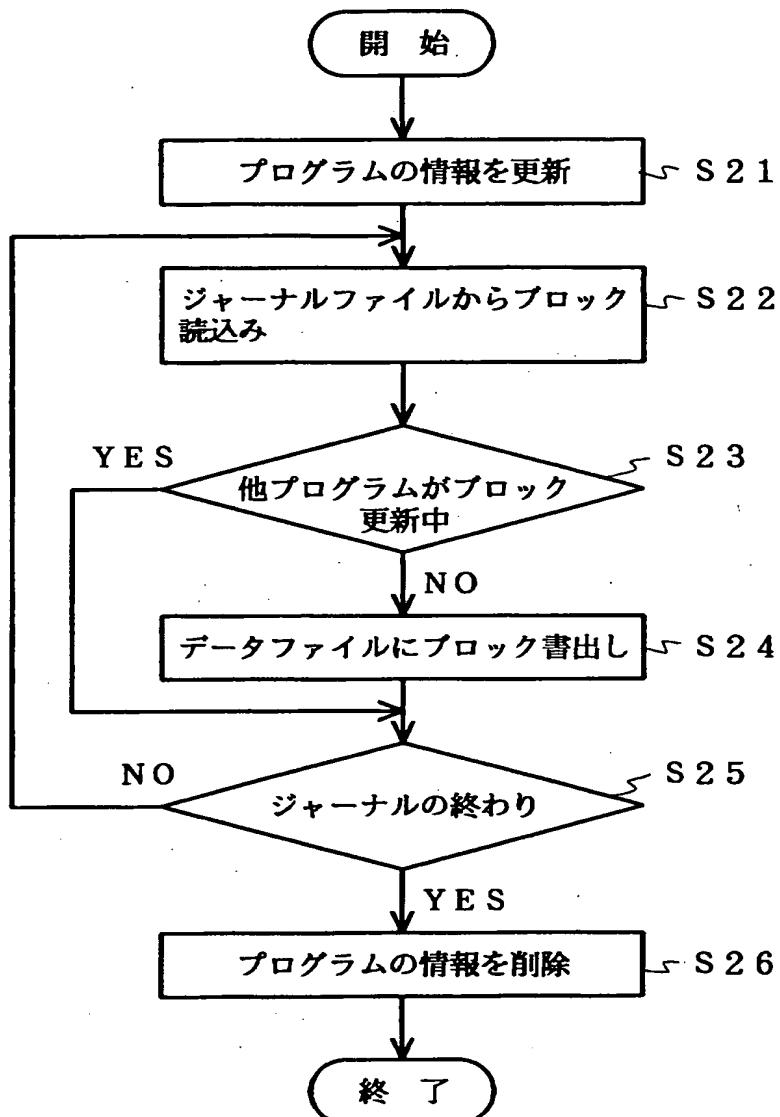
【図1】



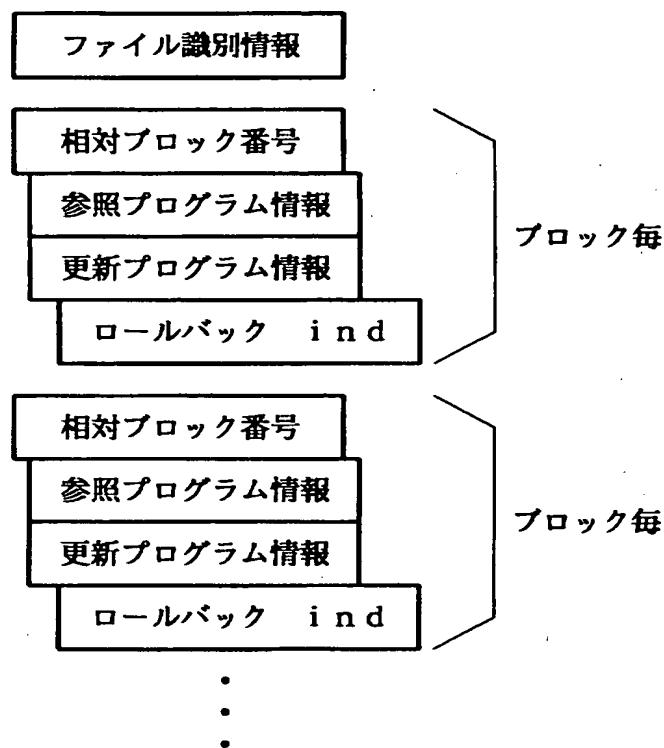
【図2】



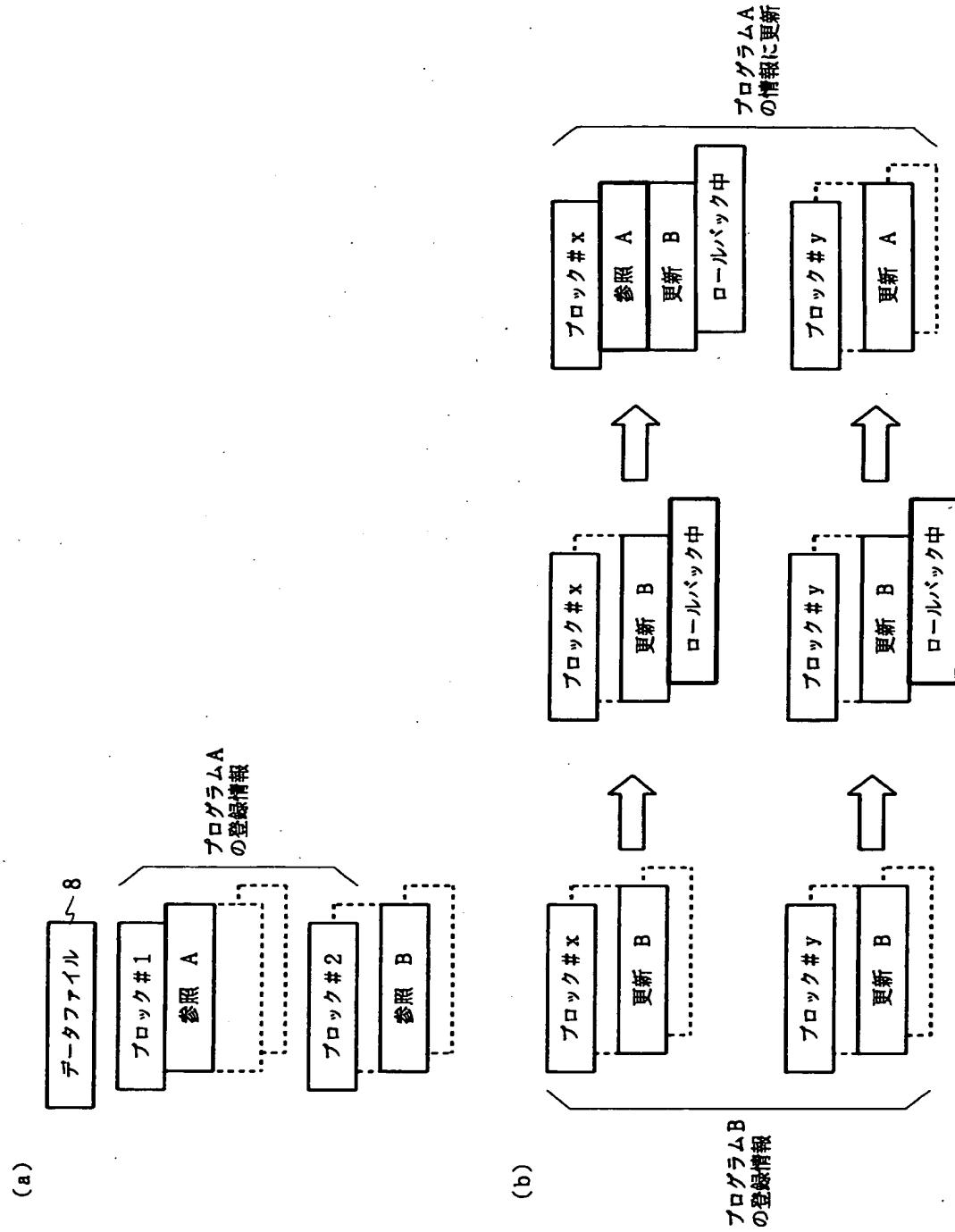
【図3】



【図4】



【図5】



【書類名】 要約書

【要約】

【課題】 ロールバックの対象となるブロックの参照・更新をその終了を待たず
に優先して行い、一つのファイルを複数のプログラムで同時に参照・更新する際
のプログラムの処理効率を向上可能なマルチプログラム処理装置を提供する。

【解決手段】 ブロックが他のプログラムで更新中であっても、ロールバック中
であれば、ファイルI/O手段1ではデータファイル8から当該ブロックを読み込
む代わりに、ジャーナル共用手段4に対して当該ブロックに係るジャーナルを要
求し、当該ブロックの内容を他のプログラムが更新する前の状態でプログラムに
渡す。プログラムからの更新要求に対しても、ファイルI/O手段1はブロック
排他手段2にブロックの状態を照会し、他のプログラムが参照も更新もしていな
ければ、ジャーナル収集手段5で更新前のブロックをデータファイル8からジャ
ーナルファイル6にジャーナルとして保存させ、データファイル8を更新する。

【選択図】 図1

出願人履歴情報

識別番号 [000004237]

1. 変更年月日 1990年 8月29日

[変更理由] 新規登録

住 所 東京都港区芝五丁目7番1号

氏 名 日本電気株式会社